Conformational Sampling with Poisson–Boltzmann Forces and a Stochastic Dynamics/Monte Carlo Method: Application to Alanine Dipeptide

JASON L. SMART, TAMI J. MARRONE, J. ANDREW McCAMMON

Departments of Chemistry and Biochemistry, and Pharmacology, University of California at San Diego, La Jolla, California 92093-0365

Received 7 November 1996; accepted 23 April 1997

ABSTRACT: We apply a combination of stochastic dynamics and Monte Carlo methods (MC/SD) to alanine dipeptide, with solvation forces derived from a Poisson–Boltzmann model supplemented with apolar terms. Our purpose is to study the effects of the model parameters, such as the friction constant and the size of the electrostatic finite difference grid, on the rate of conformational sampling and on the accuracy of the resulting free energy map. For dialanine, a converged Ramachandran map is produced in significantly less time than what is required by stochastic dynamics or Monte Carlo alone. MC/SD is also shown to be faster, per timestep, than explicit methods. © 1997 John Wiley & Sons, Inc. *J Comput Chem* **18**: 1750–1759, 1997

Keywords: Langevin dynamics; Monte Carlo; Poisson–Boltzmann equation; alanine dipeptide; continuum solvent

Correspondence to: J. L. Smart
Contract/grant sponsor: National Science Foundation
Contract/grant sponsor: National Institutes of Health
Contract/grant sponsor: NSF Supercomputer Centers

CCC 0192-8651 / 97 / 141750-10

Introduction

Stochastic dynamics simulations are based on the Langevin equation, which models dynamic solvent effects using two force terms, a stochastic force and a friction force. These forces, together with the use of a solvent-averaged potential of mean force, effectively replace, with a continuum model of solvent, the hundreds or thousands of explicit water molecules typically included in a molecular dynamics simulation. Although all explicit detail of the motions and distributions of solvent molecules is lost, the thermally averaged distribution of solute conformations is preserved. Proper choice of the friction force can also yield reasonably accurate dynamics for the solute.

In recent years stochastic dynamics has found an increasing number of biological applications, including the study of membrane phospholipids[1–3] and macromolecular crowding effects in cytoplasm.[4] A number of studies have compared stochastic dynamics to molecular dynamics[5,6] or experimental techniques.[7] These studies found Langevin dynamics to be in accord with other computational and experimental methods, provided the correct friction constant was chosen. Additional research has focused on enhancing the rate of conformational sampling of stochastic dynamics.[8–11] For a good review of Langevin dynamics applied to biological systems, with an emphasis on membranes, see Ref. 1.

To improve conformational sampling rates in dynamic simulations, stochastic dynamics and Monte Carlo methods may be combined to form a hybrid sampling algorithm, which we call MC/SD. With Monte Carlo included, dynamic information is lost, but the equilibrium properties of the solute are preserved. The implementation of MC/SD methods is straightforward, because Monte Carlo and stochastic dynamics both sample from a canonical Boltzmann-weighted distribution, and because the continuum solvent model is easily managed during Monte Carlo dihedral rotations (unlike explicit simulations, where Monte Carlo steps require the displacement of large numbers of water molecules). The MC/SD technique was first used by Guarnieri and Still[12] and later by McDonald and Still,[13] and Williams and Hall.[14] For *n*-pentane, Guarnieri and Still observed a significant improvement in the sampling rate over that of Monte Carlo or stochastic dynamics alone. Related techniques include a hybrid molecular dynamics/Monte Carlo method,[15] where the velocities are randomly reassigned after a specified number of dynamics steps, and an enhanced MC/SD implementation that uses previous knowledge of energy minima to bias the Monte Carlo sampling.[16]

In the present work, we combine the sampling advantages of MC/SD methods with the accuracy afforded by the use of new solvent-averaged force expressions derived from Poisson–Boltzmann models (see below). We used the blocked alanine molecule (*N*-acetylalanine *N*′-methylamide), commonly known as alanine dipeptide. Although the alanine dipeptide molecule has only 22 atoms, it contains, within the two backbone dihedrals ϕ and ψ, much of the conformation flexibility found in larger peptides and proteins. It is therefore an ideal test case for new dynamics methods and conformational search tools. The dipeptide has been studied extensively using molecular dynamics-based techniques,[17–19] integral equation methods,[20,21] and a host of other techniques.[8,11,22–26] For a recent review of alanine dipeptide simulations, see Ref. 27.

The primary advantage of using Langevin dynamics with a continuum treatment of solvent over traditional molecular dynamics is the time savings afforded by replacing explicit water molecules with the continuum. However, the time saved is strongly dependent on the continuum parameters, such as the size of the electrostatic grid for Poisson–Boltzmann models, and the value of the Langevin friction constant. After a brief discussion of the methods involved, therefore, this article explores the effects of variations of the continuum parameters on the speed and accuracy of the method. We then compare the MC/SD free energy map to Ramachandran maps produced using other published methods. Afterward, we present a Ramachandran map generated without the benefit of Monte Carlo sampling, to demonstrate the added efficiency gained by including occasional Monte Carlo steps. Finally, we compare the MC/SD method to standard molecular dynamics to assess the time saved by using a continuum approach.

Methods

For the present simulations, it was assumed that solvent relaxation is instantaneous and that hydrodynamic interaction effects acting on the so-

lute are negligible. The generalized Langevin equation then reduces to the simpler form:[28,29]

$$\frac{d\mathbf{v}}{dt} = -\gamma\mathbf{v} + \frac{1}{m}\mathbf{A}(t) + \frac{1}{m}\mathbf{F}_{ext}$$

where $\mathbf{v}$ is the velocity of a given solute atom, γ is the friction constant, and $\mathbf{A(t)}$ is the stochastic force. All remaining forces acting on the solute are contained in $\mathbf{F}_{ext}$. A discussion of each follows.

Solute–solvent electrostatic interactions are determined from a finite difference solution of the Poisson–Boltzmann equation. The resulting electrostatic forces are represented by three terms—a $q\mathbf{E}$ term, which contains reaction field interactions; a dielectric pressure term; and an ionic pressure term.[30]

In our model, the apolar solvation force is obtained from the gradient of the apolar energy, which is linearly related to the solvent-accessible surface area. The linear relationship between surface area and hydrocarbon transfer free energy has been observed in solubility and free energy studies of small alkanes.[31–33] Previous work[32] has shown that, using a proportionality constant of 6 cal/mol/Å^2, multiplied by the solvent accessible surface area of the solute, works well to describe the apolar solvation energy of small molecules in continuum. Thus, we have chosen this value to describe the apolar solvation energy in our simulations.

Additionally, $\mathbf{F}_{ext}$ includes intermolecular and solute–solute Lennard–Jones and Coulombic interactions, as well as bonded interactions (bond stretching, angle bending, and dihedral rotation). For a detailed description of these forces, see Ref. 11.

The Langevin equation was implemented in the program UHBD,[34] using the van Gunsteren and Berendsen leap-frog algorithm for stochastic dynamics.[35] Our alanine dipeptide model was built with Quanta 4.1[36] and used CHARMm/MSI bonded and nonbonded parameters.[36] The stepsize for all simulations was 1 fs. The surface areas[37] and derivatives[38] for the apolar solvation term were calculated using numerical methods.

In Langevin's equation, the friction term depends on the friction constant, γ, and therefore on the solvent viscosity. However, the stochastic term in Langevin's equation also depends on the friction constant, through the fluctuation–dissipation theorem.[28] Therefore, choosing the value of γ that maximizes conformational sampling is a balancing act between reducing solvent viscosity and increasing thermal fluctuations. It is also necessary to keep the friction large enough to maintain reasonably ergodic behavior of the solute. Our initial simulations found that a friction constant between 2 ps^{-1} and 10 ps^{-1} maximized conformational sampling, although the smaller values led to greater instability in the calculations. We therefore set the friction constant to 10 ps^{-1} for all simulations discussed in what follows.

It should be noted that our friction constant is smaller than that predicted by Stokes' law for a small solute in water (50 ps^{-1}). Thus, the motions of solute atoms will be more inertial in our simulations, and dynamic information, such as the interconversion rates between minima, will be inaccurate. However, the free energy surface produced from our simulations will be the same no matter what friction constant we have selected, provided γ is not too small.

In initial calculations, we found Langevin dynamics insufficient for sampling all solute conformations. It was therefore necessary to complement the stochastic dynamics simulations with occasional Monte Carlo steps, allowing the dipeptide to rapidly cross large energy barriers. One Monte Carlo step was done for every 100 stochastic dynamics steps. During the Monte Carlo step, either the ϕ or ψ backbone dihedral was randomly rotated to a new position, and the new conformation was accepted/rejected based on the Metropolis criteria.[39] If the new conformation was accepted, atomic velocities were reassigned based on a Maxwell–Boltzmann distribution for the simulation temperature (300 K), and the process was repeated. The Monte Carlo energy calculations always used the same finite difference grid as the stochastic dynamics electrostatic force calculations.

The first MC/SD simulation discussed below used conservative settings for all parameters, providing a benchmark to which all later simulations could be compared. The benchmark simulation was 2 ns in length, with a 10 ps^{-1} friction constant. The Poisson–Boltzmann finite difference grid was $30 \times 30 \times 30$, with a 0.6-Å spacing. The modest size of the grid, a necessity for achieving convergence in a reasonable amount of computer time, was compensated for by the use of Debye–Huckel boundary conditions at the grid boundaries.[34] The apolar, van der Waals and bonded forces were updated at every dynamics step. The electrostatic forces, which took considerably longer to calculate, were updated only after an atom moved more than 0.3 Å from its previous location. A separate calculation for a restricted region of configuration

space ($-180 < \phi < 0$) showed good agreement of results for simulations in which electrostatic forces were updated in this way, or after every step.

Results

The first two figures were generated from the 2-ns MC/SD benchmark simulation. For comparison, the Ramachandran map is displayed both as a free energy (G) surface (Fig. 1) and as a probability map (Fig. 2). Unlike probability maps, free energy surface display greatly exaggerated features in unsampled regions (where $G = -kT \ln \rho$ diverges). Therefore, most subsequent Ramachandran maps are displayed as probabilities rather than free energies.

To study convergence of the benchmark simulation, relative changes in the free energy surface were measured using:

$$(\Delta G)^2 = \int d\phi \int d\psi (G_{i+1} - G_i)^2 \, exp(-G_{i+1}/kT)$$

$$= \int d\phi \int d\psi (kT \, ln(\rho_i/\rho_{i+1}))^2 \rho_{i+1}$$

where ρ_{i+1} and ρ_i are the cumulative probabilities at times i and $i + 1$ (as functions of ϕ and ψ). The scalar quantity, $(\Delta G)^2$, is the sum over the full ϕ, ψ surface of the Boltzmann-weighted difference between two Ramachandran maps. Boltzmann-weighting ensures that regions of high energy will not contribute greatly to the overall difference between the two Ramachandran maps, a particularly important feature when one is interested primarily in the relative free energy of thermally accessible regions.

Figure 3 plots the difference, $(\Delta G)^2$, in cumulative Ramachandran maps generated at 50-ps inter-

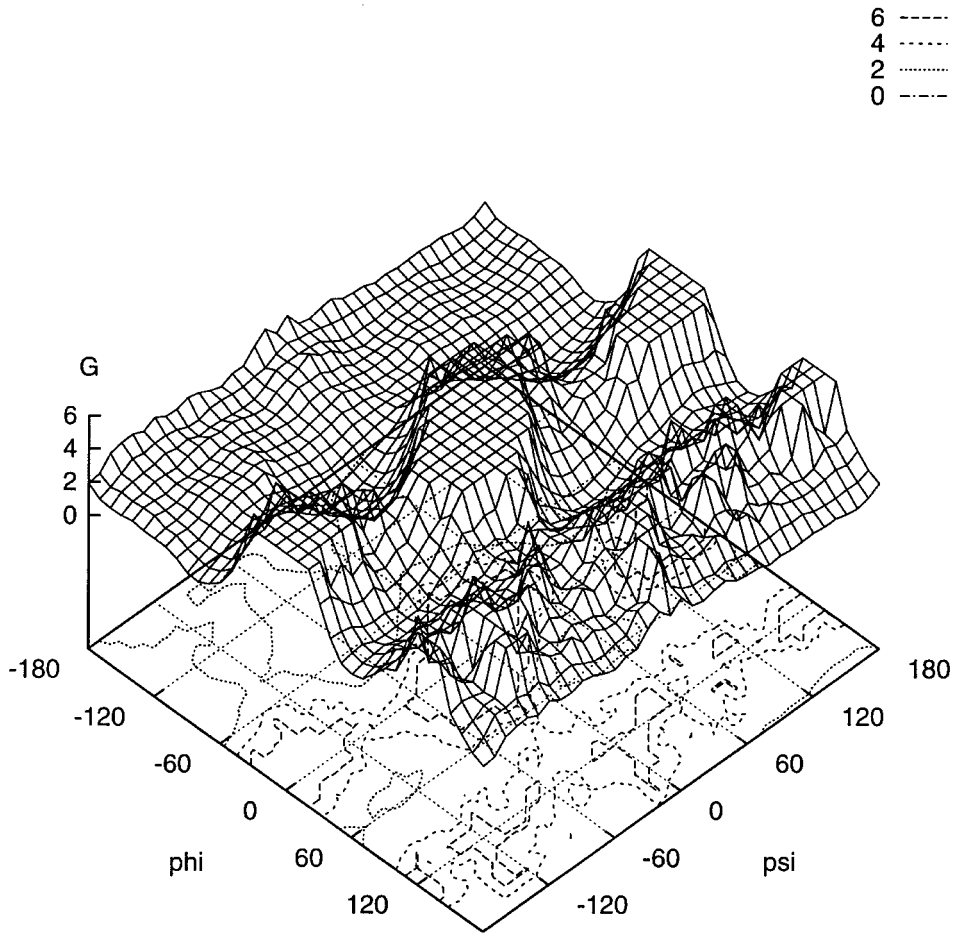

FIGURE 1. Benchmark alanine dipeptide free energy map (kcal/mol).

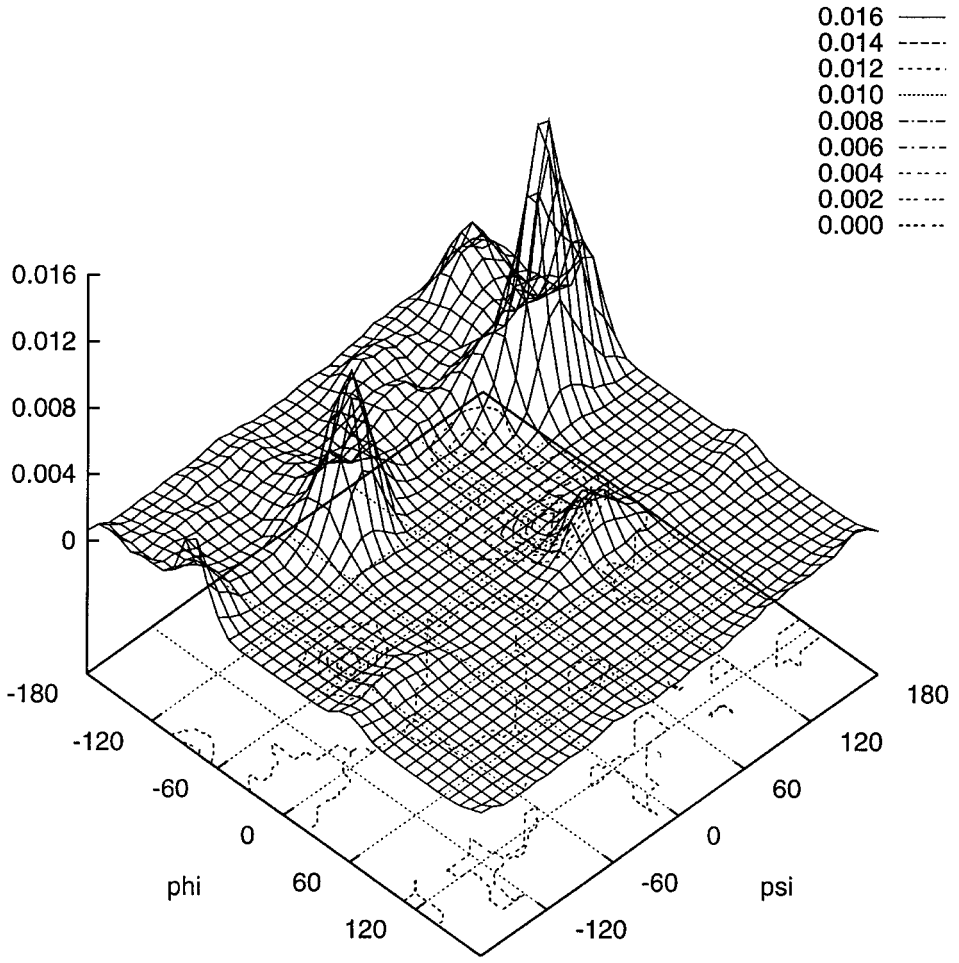

FIGURE 2. Benchmark alanine dipeptide probability map. Three-dimensional surface shown for added clarity. Units are deg^{-2}.

vals during the benchmark run. After 500 ps, the difference in successive maps approaches zero, suggesting that the benchmark simulation is converged. There is still the possibility that a region of the free energy surface has remained unexplored —the above convergence test only measures differences in the sampling of free energy surfaces, not overall convergence. However, this situation is unlikely because nearly all regions of ϕ, ψ space have been visited after 500 ps (due to the Monte Carlo sampling), and all minima have likely been found.

Convergence of the benchmark simulation may also be examined using the Boltzmann-weighted difference map, defined as:

$$\Delta G(\phi, \psi) = (G_{i+1} - G_i) exp(-G_{i+1}/kT)$$
$$= kT\, ln(\rho_i/\rho_{i+1}))\rho_{i+1}$$

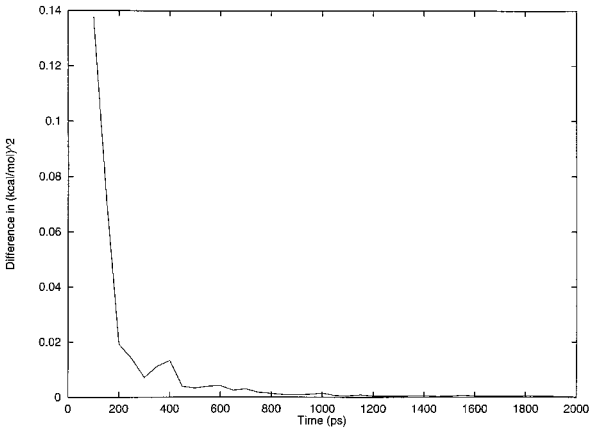

FIGURE 3. Boltzmann-weighted difference, $(\Delta G)^2$, between cumulative Ramachandran maps at 50-ps intervals for the benchmark run (kcal/mol)^2.

again where ρ_i and ρ_{i+1} are probabilities at times i and $i + 1$, as functions of ϕ and ψ. In the present case, ρ_i represents the benchmark map after 500 ps, and ρ_{i+1} represents the map after a full 2 ns. The difference map (Fig. 4) demonstrates the close agreement between the 500-ps and 2-ns results—they differ by 0.001 kcal/mol or less over the full surface. Therefore, all changes in the surface beyond the first 500 ps are minimal, and convergence is likely.

One approximation already built into the benchmark run is the 0.3-Å update criterion for electrostatic forces. The justification for this approximation comes from analysis of force autocorrelations, which quantify the rate of change of the forces. The autocorrelations are defined as:

$$C(\Delta t) = \langle \mathbf{F}(t + \Delta t) \cdot \mathbf{F}(t) \rangle_t$$

where $\mathbf{F}$ represents the Poisson–Boltzmann (PB) forces.

The decay time of $C(t)$ provides insights into allowable Poisson–Boltzmann update frequencies. If $C(t)$ decays rapidly, the PB forces applied to an atom change in magnitude or direction over very short times. If $C(t)$ exhibits a long decay time, the forces acting on solute atoms change very slowly, and may be updated less frequently.

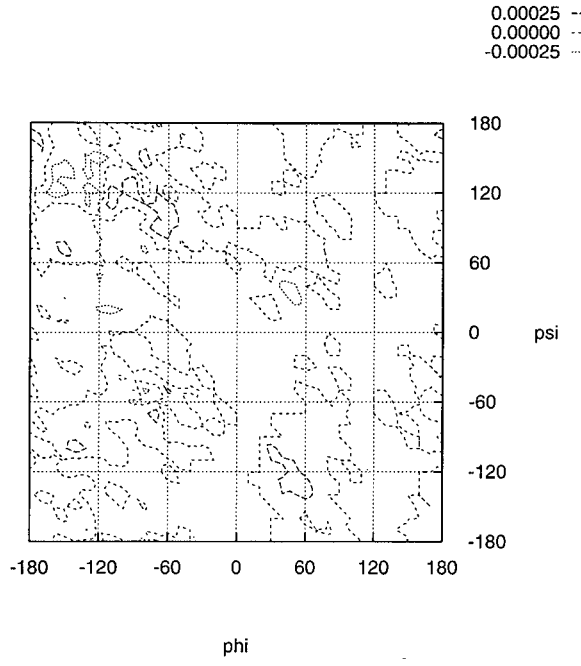

FIGURE 4. Boltzmann-weighted difference map between the first 500 ps and the full 2 ns of the benchmark run (kcal/mol).

The Poisson–Boltzmann forces are due to electrostatic interactions between the solute and the surrounding solvent. Therefore, PB forces are strongest for solvent-exposed or highly charged solute atoms. The correlation times for these atoms (defined as the time over which the correlation function falls to 1/e of its initial value) are typically around 1500 fs at 300 K. The remaining atoms, which feel only weak PB forces, exhibit two correlation times—an extremely short time due to displacements arising from the stochastic force coupled with viscous drag, and a long correlation time identical to that of the charged/exposed atoms (see Fig. 5). Because the short correlation time is of the order of a single timestep, accurate representation of the Poisson–Boltzmann forces for these atoms would require updating the forces at every step. However, because the PB forces acting on these atoms are weak, they do not contribute strongly to the overall dynamics of the solute. Thus, in the interests of reducing computational demand, it was possible to update PB forces less frequently without introducing excessive error. As an update criterion, we chose to recalculate Poisson–Boltzmann forces only when an atom moved more than 0.3 Å from its location when the forces were last calculated. This led to updates of the PB forces after about every 7–10 dynamics steps (7–10 fs), still significantly shorter than the long correlation times.

Once convergence of the benchmark run was established, it was then possible to compare it to faster runs, utilizing coarser approximations for the model parameters. Because the largest compu-

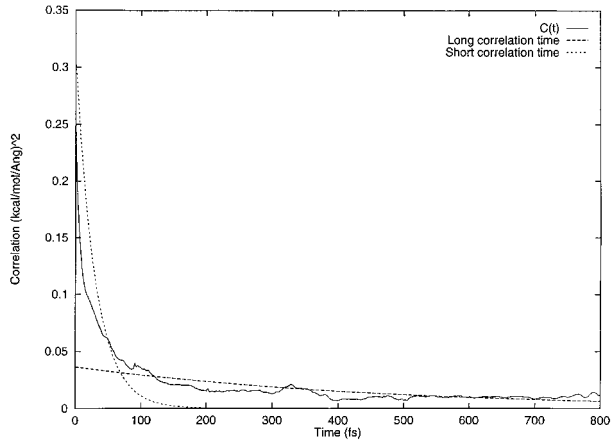

FIGURE 5. Autocorrelation function for Poisson–Boltzmann forces, $C(t)$, for a methyl hydrogen in alanine dipeptide. Exponential fits using short and long correlation times are displayed.

tational bottlenecks are the Poisson–Boltzmann electrostatics calculations, the electrostatic parameters are the most critical for reducing computation times. Above, the 0.3-Å update criterion for the Poisson–Boltzmann forces was already discussed. Another critical parameter is the size of the finite difference grid. Figure 6 was generated using a $15 \times 15 \times 15$ grid with a 1.2-Å spacing (compared to the $30 \times 30 \times 30$, 0.6-Å benchmark map). The resulting coarse grid probability map is similar to the benchmark map, although it contains a shorter and wider peak at the β conformation ($\phi = -70, \psi = 140$). The greater flexibility of the dipeptide exhibited at β is likely due to larger fluctuations in the Poisson–Boltzmann forces, an artifact of the coarser grid. However, the exponential dependence of the probability on free energy exaggerates the differences between the simulations: the Boltzmann-weighted difference map is never larger than 0.01 kcal/mol (Fig. 7), suggesting that the two simulations give approximately the same result.

One can further establish the reasonable agreement of the two simulations by comparing the relative energies of different dipeptide conformations. Table I lists the free energies of prominent minima on the dipeptide free energy surface. The energies were taken from the deepest point of each minima. From the table it is clear that the agreement between the two simulations is fairly strong —they never differ by more than about 0.2 kcal/mol. Therefore, the approximation of using a grid with a 1.2-Å spacing leads to only modest errors in the free energy surface.

Using a coarser grid, however, does lead to a significant reduction in computation time. The benchmark run was performed on the Cray C90 at

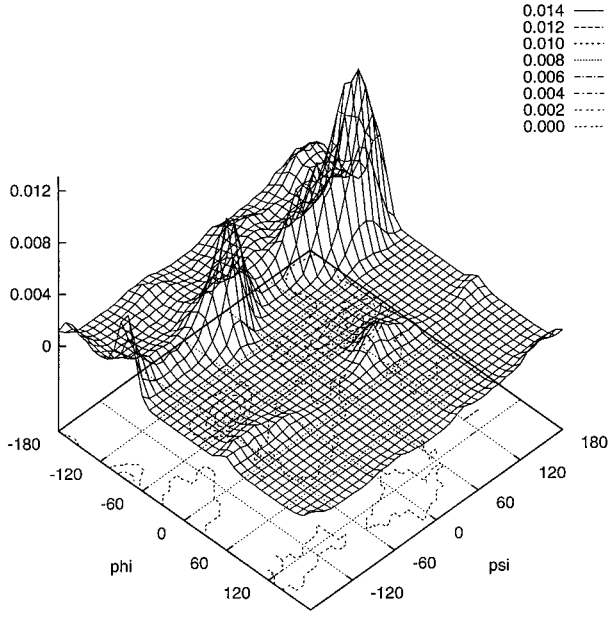

FIGURE 6. Alanine dipeptide probability Ramachandran map produced with a coarse grid (15^3, 1.2-Å spacing). Units are deg^{-2}.

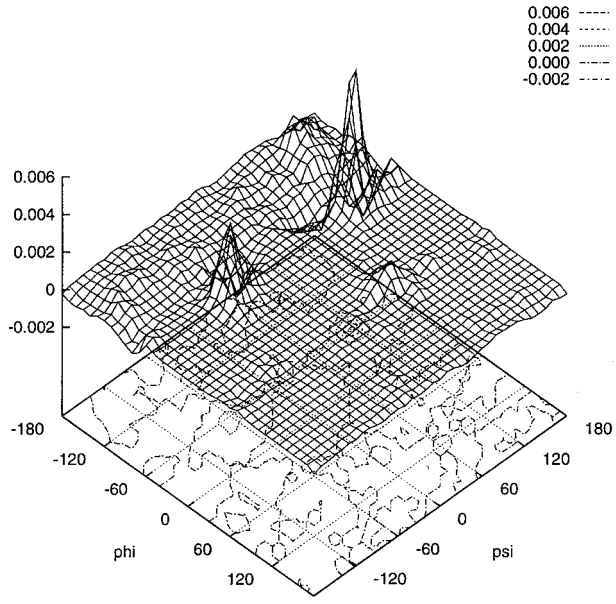

FIGURE 7. Boltzmann-weighted difference map between coarse grid (Fig. 6) and benchmark run (kcal/mol).

TABLE I.
Comparison of Free Energies between Benchmark and Coarse Grid Simulations on Alanine Dipeptide (Kilocalories per Mole). The Location of Minima Are Specified Just Below the Energies, for Reference.

Simulation	β	α_R	α_L	C_5	C_7ax
Coarse grid	0.0	0.22	1.1	0.82	1.65
	(−70, 140)	(−70, −50)	(50, 40)	(−160, 150)	(50, −130)
Benchmark	0.0	0.32	0.98	0.75	1.77
	(−70, 120)	(−70, −50)	(50, 50)	(−160, 150)	(50, −140)

the San Diego Supercomputer Center. Rough performance calculations showed that our implementation of UHBD on the Cray is about nine times faster than UHBD on an SGI R4400. The Cray benchmark run thus took about 0.7 SGI second per timestep, whereas the coarse grid simulation on the SGI took 0.1 second per timestep. If the vast majority of the simulation time was spent performing the electrostatics calculations, then an eightfold reduction in the overall grid size (from 30^3 to 15^3) would result in about an eightfold reduction in computation time. The sevenfold reduction in speed actually observed therefore demonstrates the strong dependence on electrostatics.

Although the coarse grid simulation is in reasonable agreement with the benchmark run, a stricter test of its validity is a direct comparison with other published techniques. Table II is a summary of published free energy results for the alanine dipeptide. The predicted conformational energies vary by up to about 3 kcal/mol, due to differences in parameters and in the methods themselves. However, the results of the different methods, including our MC/SD implementation, are in rough accord. For a related table, comparing solvation energies of dipeptide conformations, see Table II of Ref. 27.

The coarse grid simulation was also performed with a reduced friction constant of 4 ps^{-1}. The smaller friction constant led to greater mobility of the solute, but the improvement in conformational sampling was small. It was therefore decided that, with frequency Monte Carlo steps, the added sampling benefits of a reduced friction constant were largely unnecessary, and the values of 10 ps^{-1} was used for all remaining calculations.

Because the alanine dipeptide's surface area remains roughly constant as the molecule adopts new conformations, the relative contributions of the apolar forces are typically small. We found that applying the 0.3-Å update criteria to the apolar forces resulted in a nearly identical Ramachandran map to that generated using updates at every step. The apolar forces are likely to be more significant for larger peptides, where conformational shifts are accompanied by greater changes in the solvent-exposed surface area.

The added benefit of Monte Carlo sampling becomes evident when the above benchmark run is compared to a simulation generated with Langevin dynamics alone (Fig. 8). The benchmark run converges after only 500 ps, but without Monte Carlo, Langevin dynamics fails to converge after a full 2 ns. In general, the simple stochastic dynamics algorithm has trouble crossing barriers larger than a few kilocalories per mole, and tends to become trapped in local minima. As in the case of the simulation depicted in Figure 8, this leads to

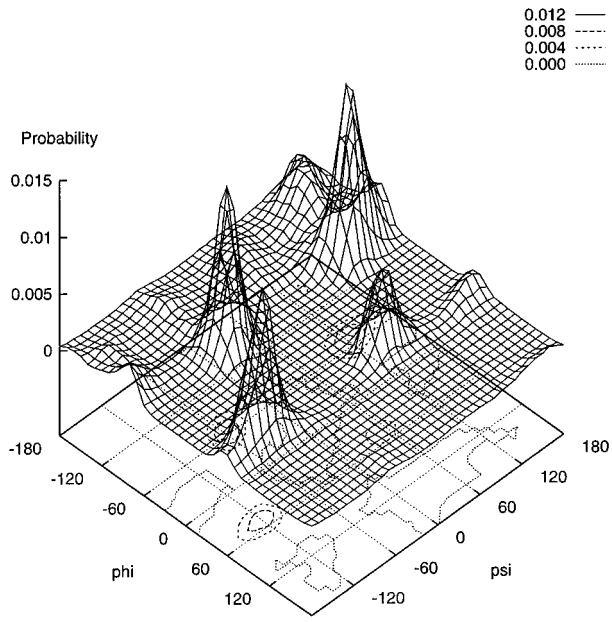

FIGURE 8. Alanine dipeptide probability Ramachandran map generated without Monte Carlo.

TABLE II.
Comparison of Free Energies Between Coarse Grid Simulation and Several Published Techniques (Kilocalories per Mole).

Simulation	β (−70, 120)	α_R (−70, −50)	α_L (50, 50)	C_5 (−160, 150)	C_7ax (50, −140)
Coarse grid	0.0	0.22	1.1	0.82	1.65
Tobias and Brooks[18]	0.0	0.2	4.1	—	3.6
Anderson and Hermans[19]	0.0	1.5	2.5	—	2.7
Pettitt and Karplus[21]	0.0	1.8	1.1	0.1	0.5
Schmidt and Fine[23]	0.0	1.3	2.0	0.4	3.6

sharp peaks about minima with little sampling elsewhere.

After verifying the accuracy of our MC/SD implementation, it was important to establish whether our simulations were indeed faster than analogous molecular dynamics calculations with explicit solvent. We therefore performed an explicit CHARMM[40] MD run on the alanine dipeptide and compared the time required for a dynamics step to the MC/SD method. The dialanine was placed in a 20-Å box and solvated with 252 TIP3P waters.[41] Nonbonded pair interactions were updated every 10 steps, and nonbonded cutoffs were truncated at 10 Å. Before dynamics, the minimized alanine–water system was heated for 200 steps (0.2 ps) and then equilibrated for 2000 steps (2 ps). All runs were performed on a Silicon Graphics Indigo2 Extreme workstation, running at 150 MHz.

The explicit molecular dynamics simulation took 1.1 seconds per dynamics step, whereas the coarse grid MC/SD simulation took 0.1 second. Thus, timestep-for-timestep, the stochastic dynamics implementation was about ten times faster. This speedup is dependent on various factors, including the update frequency of nonbonded lists, the size of nonbonded cutoffs, and the update frequency of the electrostatic forces in the continuum calculation. For the comparisons above, nonbonded pair interactions were updated every 10 steps, whereas the 0.3-Å update criteria for electrostatic forces led to their recalculation after about every 7 MC/SD timesteps. Note that with Monte Carlo sampling and a small friction constant, the MC/SD implementation requires fewer timesteps to converge.

Conclusions

The continuum solvation model used in this study has demonstrated its accuracy and speed when applied to the alanine dipeptide molecule. Certainly, the greatest disadvantage of using a continuum solvent is the loss of all solvent structural information. For example, hydrogen-bonding interactions, which are significant for some systems, cannot be modeled in a detailed way in a continuum environment. However, water molecules playing key structural or functional roles may be explicitly included in the calculations at little expense to overall performance.[5] Therefore, many of the limitations of the continuum solvent model may be partially overcome with only slight modifications.

Fortunately, the lack of explicit waters leads to a straightforward implementation of Monte Carlo methods; and Monte Carlo methods considerably enhance the rate of conformational sampling. When Monte Carlo methods are applied to larger systems, however, additional difficulties arise. For larger molecules (especially proteins), random conformational sampling often leads to huge overlaps between subdomains and the subsequent rejection of most MC steps. To improve acceptance rates, energy minimization may be employed before applying the Metropolis criterion. There have been many conformational studies in the literature using combined Monte Carlo/energy minimization techniques.[42–45] The inclusion of MC/EM in dynamics simulations is a relatively new approach, however, and will likely be employed in our future work on larger systems.

The speed of the Monte Carlo/stochastic dynamics implementation is currently limited by the electrostatics. Additional modifications will therefore include faster techniques for solving the Poisson–Boltzmann equation, such as multigrid[46,47] or boundary-element methods.[48] Faster techniques for exploring conformation space are also a possibility. Biasing potentials, which artificially reduce the size of energy barriers during a simulation, have been successfully applied to a variety of small-molecule systems,[17,49–52] and will likely be added to our MC/SD implementation in the future.

The MC/SD simulations, although faster than explicit methods, still give results in accord with conservative continuum calculations and with other published techniques. The approximations introduced in the model are therefore justified for the simple case of alanine dipeptide. Certainly, for larger peptides, and for proteins, additional complications will arise. However, it is likely that most of the limitations of the model can be overcome with simple modifications or additions to the MC/SD implementation.

Acknowledgments

We are very grateful for advice and software help from Dr. Jim Briggs, Dr. Adrian Elcock, Dr. Michael Gilson, and Michael Potter. We also thank Dr. S. Sridharan for providing us with the code to calculate surface areas and surface area derivatives. J. S. is the recipient of a stipend from the

NIH/UCSD Molecular Biophysics Training Grant. T. J. M. was supported in part by an NIH postdoctoral fellowship.

References

1. R. W. Pastor, In *The Molecular Dynamics of Liquid Crystals*, G. R. Luckhurst, and C. A. Veracini, Eds., Kluwer, Dordrecht, 1994, p. 85.
2. L. L. Pearce and S. C. Harvey, *Biophys. J.*, **65**, 1084 (1993).
3. P. H. Konstant, L. L. Pearce, and S. C. Harvey, *Biophys. J.*, **67**, 713 (1994).
4. D. J. Bicout and M. J. Field, *J. Chem. Phys.*, **100**, 2489 (1996).
5. G. Widmalm and R. W. Pastor, *J. Chem. Soc. Faraday Trans.*, **88**, 1747 (1992).
6. M. Depner, B. L. Schurmann, and F. Auriemma, *Mol. Phys.*, **74** 715 (1991).
7. F. Liu, W. J. Horton, C. L. Mayne, T. Xiang, and D. M. Grant, *J. Am. Chem. Soc.*, **114**, 5281 (1992).
8. R. J. Loncharich, B. B. Brooks, and R. W. Pastor, *Biopolymers*, **32**, 523 (1992).
9. B. Cartling, *J. Chem. Phys.*, **91**, 427 (1989).
10. N. Gronbech-Jensen and S. Doniach, *J. Comput. Chem.*, **15**, 997 (1994).
11. M. K. Gilson, J. A. McCammon, and J. D. Madura, *J. Comput. Chem.*, **16**, 1081−1095 (1995).
12. F. Guarnieri and W. C. Still, *J. Comput. Chem.*, **15**, 1302 (1994).
13. D. Q. McDonald and W. C. Still, *J. Am. Chem. Soc.*, **116**, 11550 (1994).
14. D. J. Williams and K. B. Hall, *J. Phys. Chem.*, **100**, 8224 (1996).
15. A. Brass, B. J. Pendleton, Y. Chen, and B. Robson, *Biopolymers*, **33**, 1307 (1993).
16. H. Senderowitz, F. Guarnieri, and W. C. Still, *J. Am. Chem. Soc.*, **117**, 8211 (1995).
17. T. P. Straatsma and J. A. McCammon, *J. Chem. Phys.*, **101**, 5032 (1994).
18. D. J. Tobias and C. L. Brooks, *J. Phys. Chem.* **96**, 3864 (1992).
19. A. G. Anderson and J. Hermans, *Proteins: Struct. Funct. Genet.* **3**, 262 (1988).
20. B. M. Pettitt, M. Karplus, and P. J. Rossky, *J. Phys. Chem.*, **90**, 6335 (1986).
21. B. M. Pettitt and M. Karplus, *J. Phys. Chem.*, **92**, 3994 (1988).
22. T. J. Marrone, M. K. Gilson, and J. A. McCammon, *J. Phys. Chem.*, **100**, 1439 (1996).
23. A. B. Schmidt and R. M. Fine, *Mol. Simul.*, **13**, 347 (1994).
24. P. E. Smith, B. M. Pettitt, and M. Karplus, *J. Phys. Chem.*, **97**, 6907 (1993).
25. M. Pellegrini, N. Gronbech-Jensen, and S. Doniach, *J. Chem. Phys.*, **104**, 8639 (1996).
26. S. Kumar, P. W. Payne, and M. Vasquez, *J. Comput. Chem.*, **17**, 1269 (1996).
27. C. L. Brooks and D. A. Case, *Chem. Rev.*, **93**, 2487 (1993).
28. H. L. Friedman, *A Course in Statistical Mechanics*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
29. S. Chandrasekhar, *Rev. Mod. Phys.*, **15**, 1 (1943).
30. M. K. Gilson, M. E. Davis, B. A. Luty, and J. A. McCammon, *J. Phys. Chem.*, **97**, 3591 (1993).
31. R. B. Hermann, *J. Phys. Chem.*, **76**, 2754 (1972).
32. T. Simonson and A. T. Brunger, *J. Phys. Chem.*, **98** 4683 (1994).
33. D. Sitkoff, K. A. Sharp, and B. Honig, *J. Phys. Chem.*, **98**, 1978 (1994).
34. M. E. Davis, J. D. Madura, B. A. Luty, and J. A. McCammon, *Comput. Phys. Commun.*, **62**, 187 (1991).
35. W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Simul.*, **1**, 173 (1988).
36. *Quanta 4.1*, Molecular Simulations, Inc., San Diego, CA.
37. S. Sridharan, A. Nicholls, and B. Honig, *Biophys. J.*, **61**, A174 (1992).
38. S. Sridharan, A. Nicholls, and K. A. Sharp, *J. Comput. Chem.*, **16**, 1038 (1995).
39. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *J. Chem. Phys.*, **21**, 1087 (1953).
40. B. B. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
41. W. L. Jorgensen, J. Chandrasekhar, J. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.*, **79**, 926 (1983).
42. Z. Li and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA*, **84**, 6611 (1987).
43. A. Nayeem, J. Vila, and H. A. Scheraga, *J. Comput. Chem.*, **12**, 594 (1991).
44. W. C. Guida, R. S. Bohacek, and M. D. Erion, *J. Comput. Chem.*, **13**, 214 (1992).
45. M. Vasquez, E. Meirovitch, and H. Meirovitch, *J. Phys. Chem.*, **98**, 9380 (1994).
46. M. Holst and F. Saied, *J. Comput. Chem.*, **14**, 105 (1993).
47. M. Holst, R. E. Kozack, F. Saied, and S. Subramaniam, *Prot. Struct. Funct. Genet.*, **18**, 231 (1994).
48. R. Bharadwaj, A. Windemuth, S. Sridharan, B. Honig, and A. Nicholls, *J. Comput. Chem.*, **16**, 898 (1995).
49. T. C. Beutler, T. Bremi, R. R. Ernst, and W. F. van Gunsteren, *J. Phys. Chem.*, **100**, 2637 (1996).
50. R. W. Hooft, B. P. van Eijck, and J. Kroon, *J. Chem. Phys.*, **97**, 6690 (1992).
51. T. C. Beutler and W. F. van Gunsteren, *J. Chem. Phys.*, **100**, 1492 (1994).
52. R. K. Schmidt, B. Teo, and J. W. Brady, *J. Phys. Chem.*, **99**, 11339 (1995).